Claim Amendment Summary

Claims pending

- At time of the Action: Claims 1-33.
- After this Response: Claims 1-2 and 10-60.

Cancelled claims: 3-9.

Amended claims: 1 and 14.

New claims: 34-60.

1. **(CURRENTLY AMENDED)** A method of parsing an Extensible Markup Language (XML) data stream comprising:

defining a plurality of states, individual states being associated with individual elements of an XML data stream;

associating one or more rules with each state;

receiving an XML data stream;

evaluating the XML data stream against one or more of the rules for individual elements contained in the XML data stream; and

processing ~~disregarding~~ only those ~~associated~~ portions of the XML data stream that do not violate ~~if~~ any of the rules that are associated with those portions ~~are violated~~.

2. **(ORIGINAL)** The method of claim 1, wherein the one or more rules relate to a schema of the XML data stream.

3-9. **(CANCELLED)**

10. **(ORIGINAL)** The method of claim 1 further comprising defining one or more rules that relate to an element's contents.

11. **(ORIGINAL)** The method of claim 10, wherein said one or more rules that relate to an element's contents define which elements can be contained within other elements.

12. **(ORIGINAL)** The method of claim 11, wherein if a rule that defines which elements can be contained within other elements is violated, disregarding associated portions of the XML data stream until a close tag is received for an element that violates the rule.

13. **(ORIGINAL)** A computer-readable medium having a program thereon which, when executed by a computer, performs the steps of claim 1.

14. **(CURRENTLY AMENDED)** A method of parsing an Extensible Markup Language (XML) data stream comprising:

defining a schema module that is associated with an HTTP request type that is received from a client, the schema module having a function that determines whether an XML data stream conforms to a given schema that is associated with the HTTP request type;

evaluating an XML data stream with the schema module; and

processing ~~disregarding a~~ only those portions of the XML data stream that ~~if it does not~~ conform to the given schema.

15. **(ORIGINAL)** The method of claim 14, wherein said defining of the schema module comprises defining a plurality of schema modules, individual schema modules being associated with different HTTP request types.

16. **(ORIGINAL)** The method of claim 14, wherein said function determines whether there are any unauthorized elements that appear in a client's request.

17 **(ORIGINAL)** The method of claim 14, wherein said function determines whether there are any unauthorized elements that appear in a client's request; said disregarding comprising disregarding said XML data stream portion until a close tag is received for an unauthorized element.

18. **(ORIGINAL)** The method of claim 14, wherein said HTTP request type comprises a WebDAV request type.

19. **(ORIGINAL)** The method of claim 18, wherein said WebDAV request type comprises a PROPFIND request.

20. **(ORIGINAL)** The method of claim 18, wherein said WebDAV request type comprises a PROPPATCH request.

21. **(ORIGINAL)** The method of claim 18, wherein said WebDAV request type comprises a SEARCH request.

22. **(ORIGINAL)** The method of claim 18, wherein said WebDAV request type comprises one of a LOCK and UNLOCK request.

23. **(ORIGINAL)** A computer-readable medium having a program thereon which, when executed by a computer, performs the steps of claim 14.

24. **(ORIGINAL)** An Extensible Markup Language (XML) parsing system comprising:

a parser configured to receive an XML data stream and generate a series of calls as it parses the XML data stream;

a node factory communicatively associated with the parser and configured to receive the parser's calls and responsive thereto construct a representation of the XML data stream that the parser is parsing; and

a schema module communicatively associated with the node factory and configured to evaluate the node factory's representation of the XML data stream and determine whether it conforms to a known schema.

25. **(ORIGINAL)** The parsing system of claim 24, wherein said parsing system comprises a plurality of schema modules, each schema module being associated with a different known schema.

26. **(ORIGINAL)** The parsing system of claim 24, wherein the schema module corresponds to an HTTP request type.

27. **(ORIGINAL)** The parsing system of claim 24, wherein said parsing system comprises a plurality of schema modules, each schema module being

associated with a different known schema and corresponding to a different HTTP request type.

28. **(ORIGINAL)** The parsing system of claim 27, wherein at least one of the different HTTP request types is a WebDAV request.

29. **(ORIGINAL)** The parsing system of claim 24, wherein the schema module is configured to ignore an XML data stream portion that does not conform to the known schema.

30. **(ORIGINAL)** An Extensible Markup Language (XML) parsing system comprising:

a collection of schema modules, each of which being configured to evaluate a different schema that is associated with an XML data stream; and

a plurality of states associated with each schema module, individual states of a schema module defining a schema requirement relating to a particular element that is evaluated by that schema module.

31. **(ORIGINAL)** The parsing system of claim 30, wherein each schema module is associated with a different HTTP request and is configured to evaluate a schema that is associated with the HTTP request with which is it associated.

32. **(ORIGINAL)** The parsing system of claim 31, wherein at least one of the HTTP requests is a WebDAV request.

33. **(ORIGINAL)** The parsing system of claim 31, wherein each of the HTTP requests is a WebDAV request.

34. **(NEW)** A method of parsing an Extensible Markup Language (XML) data stream comprising:

defining a plurality of states, individual states being associated with individual elements of an XML data stream;

associating one or more rules with each state;

receiving an XML data stream;

evaluating the XML data stream against one or more of the rules for individual elements contained in the XML data stream; and

disregarding associated portions of the XML data stream if any of the rules that are associated with those portions are violated, the disregarded portions of thc XML data stream representing at least one error in the XML data stream.

35. **(NEW)** The method of claim 34, wherein the request type is a WebDAV request type.

36. **(NEW)** The method of claim 35, wherein the WebDAV request type is a PROPFIND request.

37. **(NEW)** The method of claim 35, wherein the WebDAV request type is a PROPPATCH request.

38. **(NEW)** The method of claim 35, wherein the WebDAV request type is a SEARCH request.

39. **(NEW)** The method of claim 35, wherein the WebDAV request type is one of a LOCK and UNLOCK request.

40. **(NEW)** A method of parsing an Extensible Markup Language (XML) data stream comprising:

defining a plurality of states, individual states being associated with individual elements of an XML data stream, wherein the defining of the plurality of states comprises defining one or more schema modules that are configured to track one or more states of the XML data stream;

associating one or more rules with each state;

receiving the XML data stream;

evaluating the XML data stream against one or more of the rules for individual elements contained in the XML data stream, wherein the evaluating comprises using the one or more schema modules to evaluate the XML data stream against one or more schema-based rules; and

disregarding associated portions of the XML data stream if any of the rules that are associated with those portions are violated.

41. **(NEW)** The method of claim 40, wherein each schema module is associated with at least one request type that defines the XML data stream.

42. **(NEW)** The method of claim 41, wherein the request type is a WebDAV request type.

43. **(NEW)** The method of claim 42, wherein the WebDAV request type is a PROPFIND request.

44. **(NEW)** The method of claim 42, wherein the WebDAV request type is a PROPPATCH request.

45. **(NEW)** The method of claim 42, wherein the WebDAV request type is a SEARCH request.

46. **(NEW)** The method of claim 42, wherein the WebDAV request type is one of a LOCK and UNLOCK request.

47. **(NEW)** The method of claim 40 further comprising defining one or more rules that relate to an element's contents.

48. **(NEW)** The method of claim 47, wherein said one or more rules that relate to an element's contents define which elements can be contained within other elements.

49. **(NEW)** The method of claim 48, wherein if a rule that defines which elements can be contained within other elements is violated, disregarding associated portions of the XML data stream until a close tag is received for an element that violates the rule.

50. **(NEW)** A computer-readable medium having a program thereon which, when executed by a computer, performs the steps of claim 40.

51. **(NEW)** A method of parsing an Extensible Markup Language (XML) data stream comprising:

defining a plurality of states, individual states being associated with individual elements of an XML data stream, wherein the defining of the plurality of states comprises defining one or more schema modules that are configured to track one or more states of the XML data stream each schema module being associated with at least one request type that defines the XML data stream;

associating one or more rules with each state;

receiving the XML data stream;

evaluating the XML data stream against one or more of the rules for individual elements contained in the XML data stream; and

disregarding associated portions of the XML data stream if any of the rules that are associated with those portions are violated.

52. **(NEW)** The method of claim 51, wherein the request type is a WebDAV request type.

53. **(NEW)** The method of claim 52, wherein the WebDAV request type is a PROPFIND request.

54. **(NEW)** The method of claim 52, wherein the WebDAV request type is a PROPPATCH request.

55. **(NEW)** The method of claim 52, wherein the WebDAV request type is a SEARCH request.

56. (NEW) The method of claim 52, wherein the WebDAV request type is one of a LOCK and UNLOCK request.

57. (NEW) The method of claim 51 further comprising defining one or more rules that relate to an element's contents.

58. (NEW) The method of claim 57, wherein said one or more rules that relate to an element's contents define which elements can be contained within other elements.

59. (NEW) The method of claim 58, wherein if a rule that defines which elements can be contained within other elements is violated, disregarding associated portions of the XML data stream until a close tag is received for an element that violates the rule.

60. (NEW) A computer-readable medium having a program thereon which, when executed by a computer, performs the steps of claim 51.